Unreal Engine 488,087	UEFN & Creative 9,492	Epic Games Store 318	Capturing Reality 1,989	MetaHuman 1,526	Twinmo 566

Dev Community

Join our community, grow your knowledge and learn from others!

SIGN IN

Don't have an Epic Games account?
Sign up

🏠 Home

💬 Forums

🎓 Learning

Getting Started

Epic Epic Games

👥 Community

📄 Snippets

Skip to main content

Use Emacs to develop Unreal C++ project through the help of UBT

unreal-engine

toughcactus   **Mar 24**

We could configure Unreal editor using Visual Studio or Visual Studio Code as default IDE, but they are not the only choices.

Lately I'm learning how to use Emacs, and I couldn't help to wonder if I could use Emacs to develop Unreal C++ project.

I'm not going to say I did it, but I'm okay with what I have found.

So this article is written to share how I use Emacs to develop Unreal C++ project.

I could find definitions and all the references from cpp file, it's ok for lsp to expand macros and understand the class is not a variable although the class key word followed by macro.

It's not supported to go to the implementaion from definition or declaration in the header file.

We could use completion during our development.

21/12/2023, 00:11

Use Emacs to develop Unreal C++ project through the help of UBT - Programming & Scripting / C++ - Epic Developer Community Forums

First of all, I want to introduce the packages I use.

For normal C++ project development, I use clangd + lsp-mode + company, which supports auto completion, quickly jumping to definitions and finding all the references. All it needs, it's a compile_commands.json.

For Unreal C++ project, it's so difficult to find references and definitions, due to Engine source code has 50+ G, and we use so many macros during our development.

I've tried gtags, etags, ctags, no rtags because I couldn't install it on Windows. All those tools take a lot of time to generate tags file, which didn't make it easier to find the definitions or references.

But I happened to use VSCode building Unreal C++ project, while I found it use something have common with compile_commands.json.

This is the key point, under the folder of our project named .vscode, there are 2 files, the one named compileCommands_YOURPROJECTNAME.json, could be used as compile_commands.json for project, the other one called compileCommands_Default.json is about Engine source code.

You could also find .vscode folder under "Epic Games/UE_NO/".

In order to let those files work in Emacs, I put "NO/YOURPROJECTNAME/.vscode/compileCommands_YOURPROJECTNAME.json" under "NO/YOURPROJECTNAME/" as compile_commands.json, so I could get definitions and references from my project even UE Engine Source code.

I also put "Epic Game/UE_NO/.vscode/compileCommands_Default.json" under "Epic Game/UE_NO/" as compile_commands.json, which helps us to search definitions and references in UE Engine Source folder.

It will help you to understand, that the .vscode is generated by UnrealBuildTool.exe, aimed at helping VSCode to find definitions or references, and provide other functions.

Skip to main content

It's difficult for gtags, ctags to generate tags file, I think it's because they don't know the project structure or how to parse code with so many macros.

There are 2 ways to make Unreal editor generate .vscode folder. Both of them need you set VSCode as you default IDE first.

To set VSCode as your default IDE, you need open you editor, find Editor Preferences, go to Source Code, select Visual Studio Code as you Source Code Editor.

Then you could choose the way you like to generate the .vscode folder:

- Under Tools, choose Refresh Visual Studio Code Project, then you will get the .vscode folder both under your project and the Engine.

or

- Use UnrealBuildTool.exe.
  UnrealBuildTool.exe -projectfiles -project="YOURPROJECTNAME\YOURPROJECTNAME.uproject" -game -rocket -progress

  The key parameter is -projectfiles, it tells UBT to generate project files according to your setting for default IDE.

That's all I want to share. I think it will also help if you want to use NeoVim as your C++ IDE, which supports lsp to parse source code too.

If you want to use other software to develop Unreal C++ project, I guess you may be interested in using commands building or running your game. Here is the link I found in Unreal Engine forum, which really helps me a lot.**Unreal Engine C++ Project Setup, From Scratch - YouTube**

Topic	Replies	Views	Activity

Skip to main content

Topic	Replies	Views	Activity
Using a delegate as an argument to a function (for callbacks) Unresolved **question** **CPP** **4.27+ I'm trying to implement an anonymous callback system. I'm currently using…**	0	134	**Jan 21**
Mac generate Xcode project files does nothing Unresolved **question** **Every time I try to generate Xcode project files nothing happens. Every solution…**	0	84	**Jan 27**
✓ **Local multiplayer works but not over internet despite being connect to same lobby** Resolved **question** **CPP** **onlinesubsystem** **Hi, I just finished this tutorial How To Use Sessions In C++ - UE4: Guidebook For…**	1	220	**Feb 18**
Adding to TArray in child class? Unresolved **question** **Hey guys, I am trying to make a base data asset, then have a child class with item…**	4	231	**Feb 9**
I'm trying to get a box to move and I can't see why it isn't working Unresolved **question** **Hi, I'm trying to build a game in unreal and one of the first things I'm trying to do is…**	2	167	**Jan 12**

Want to read more? Browse other topics in   or **view latest topics**.

Skip to main content

BACK TO TOP

CREATE

Fortnite Creative

Unreal Engine

Twinmotion

RealityCapture

Sketchfab

MetaHuman

Megascans

OPERATE

Game Services

Account Services

Kids Web Services

Epic Games Store

CONNECT

Epic Developer Community

ArtStation

Terms of service   **Privacy policy**   **Community Rules**